

In the Claims

This listing of claims will replace all prior versions and listings of claims in the application:

Claims 1 to 4. (Canceled)

1        5.    (Currently Amended) ~~The A processor of claim 1~~ having a  
2 changeable architected state, comprising:  
3        instruction memory for storing instructions;  
4        an instruction pipeline, wherein an instruction which passes  
5 entirely through the pipeline alters the architected state and  
6 wherein the pipeline comprises circuitry for fetching instructions  
7 from the instruction memory into the pipeline;  
8        an annul word memory for storing an annul code having a  
9 plurality of annul bits, each annul bit having a one-to-one  
10 correspondence to one instruction of a group of instructions in the  
11 pipeline; and  
12        circuitry for preventing one or more selected instructions in  
13 the group from altering the architected state in response to the  
14 corresponding annul bit of the annul code;  
15        wherein the instruction pipeline further comprises a plurality  
16 of execution units;  
17        wherein the plurality of execution units are operable such  
18 that in a given clock cycle an integer number N of the plurality of  
19 execution units are scheduled to execute;  
20        wherein the circuitry for preventing one or more selected  
21 instructions in the group from altering the architected state  
22 comprises circuitry for coupling annul bits to respective ones of  
23 the plurality of execution units; and  
24        wherein the circuitry for coupling the annul bits to  
25 respective ones of the plurality of execution units comprises  
26 circuitry for coupling only the integer number N of the annul bits

27 to the plurality of execution units which are scheduled to execute  
28 in the given clock cycle whereby on an immediately following clock  
29 cycle annul bits beginning at an N+1 annul bit are coupled to  
30 execution units scheduled to execute of that following cycle.

Claim 6. (Canceled)

1 7. (Previously Presented) The processor of claim 5:  
2 wherein the group of instructions corresponding to the annul  
3 code comprise instructions corresponding to a software loop  
4 scheduled to execute for an integer M number of iterations; and  
5 wherein during a given iteration the circuitry for preventing  
6 prevents one or more of the group of instructions corresponding to  
7 the annul bits of the annul code from altering the architected  
8 state in response to the annul bits of the annul code and the annul  
9 code based on a relationship of the given iteration to the integer  
10 M number of iterations preventing differing instructions from  
11 altering the architected state during different iterations.

Claim 8 and 9. (Canceled)

1 10. (Currently Amended) ~~The A processor of claim 1~~ having a  
2 changeable architected state, comprising:  
3 instruction memory for storing instructions;  
4 an instruction pipeline, wherein an instruction which passes  
5 entirely through the pipeline alters the architected state and  
6 wherein the pipeline comprises circuitry for fetching instructions  
7 from the instruction memory into the pipeline;  
8 an annul word memory for storing an annul code having a  
9 plurality of annul bits, each annul bit having a one-to-one  
10 correspondence to one instruction of a group of instructions in the  
11 pipeline; and

12        circuitry for preventing one or more selected instructions in  
13 the group from altering the architected state in response to the  
14 corresponding annul bit of the annul code;

15        wherein the annul code is generated in response to one or more  
16 constant generating instructions and loaded into the annul word  
17 memory.

1        11. (Currently Amended) ~~The A processor of claim 1~~ having a  
2 changeable architected state, comprising:

3        instruction memory for storing instructions;

4        an instruction pipeline, wherein an instruction which passes  
5 entirely through the pipeline alters the architected state and  
6 wherein the pipeline comprises circuitry for fetching instructions  
7 from the instruction memory into the pipeline;

8        an annul word memory for storing an annul code having a  
9 plurality of annul bits, each annul bit having a one-to-one  
10 correspondence to one instruction of a group of instructions in the  
11 pipeline; and

12        circuitry for preventing one or more selected instructions in  
13 the group from altering the architected state in response to the  
14 corresponding annul bit of the annul code;

15        wherein the annul code is loaded into the annul word memory  
16 from a memory.

1        12. (Currently Amended) ~~The A processor of claim 1~~ having a  
2 changeable architected state, comprising:

3        instruction memory for storing instructions;

4        an instruction pipeline, wherein an instruction which passes  
5 entirely through the pipeline alters the architected state and  
6 wherein the pipeline comprises circuitry for fetching instructions  
7 from the instruction memory into the pipeline;

8        an annul word memory for storing an annul code having a  
9 plurality of annul bits, each annul bit having a one-to-one  
10 correspondence to one instruction of a group of instructions in the  
11 pipeline; and

12        circuitry for preventing one or more selected instructions in  
13 the group from altering the architected state in response to the  
14 corresponding annul bit of the annul code;

15        wherein the annul code is an immediate value in an immediate  
16 operand instruction passing through the pipeline loaded into the  
17 annul word memory in response to execution of the immediate operand  
18 instruction.

Claims 13 to 15.        (Canceled)

1        16. (Currently Amended) ~~The A processor of claim 1~~ having a  
2 changeable architected state, comprising:

3        instruction memory for storing instructions;

4        an instruction pipeline, wherein an instruction which passes  
5 entirely through the pipeline alters the architected state and  
6 wherein the pipeline comprises circuitry for fetching instructions  
7 from the instruction memory into the pipeline;

8        an annul word memory for storing an annul code having a  
9 plurality of annul bits, each annul bit having a one-to-one  
10 correspondence to one instruction of a group of instructions in the  
11 pipeline; and

12        circuitry for preventing one or more selected instructions in  
13 the group from altering the architected state in response to the  
14 corresponding annul bit of the annul code;

15        wherein the annul code is loaded into the annul word memory  
16 from a selected location of memory in response to an instruction  
17 having a condition predicate;

18 wherein the annul code comprises a first annul code stored in  
19 a first location in the memory loaded into the annul word memory in  
20 response to the condition predicate being satisfied; and  
21 wherein the annul code comprises a second annul code stored in  
22 a second location in the memory loaded into the annul word memory  
23 in response to the condition predicate not being satisfied.

1 17. (Currently Amended) ~~The A processor of claim 1~~ having a  
2 changeable architected state, comprising:  
3 instruction memory for storing instructions;  
4 an instruction pipeline, wherein an instruction which passes  
5 entirely through the pipeline alters the architected state and  
6 wherein the pipeline comprises circuitry for fetching instructions  
7 from the instruction memory into the pipeline;  
8 an annul word memory for storing an annul code having a  
9 plurality of annul bits, each annul bit having a one-to-one  
10 correspondence to one instruction of a group of instructions in the  
11 pipeline; and  
12 circuitry for preventing one or more selected instructions in  
13 the group from altering the architected state in response to the  
14 corresponding annul bit of the annul code;  
15 and further comprising a first data register and a second data  
16 register;  
17 wherein the annul code is loaded into the annul word memory  
18 from a selected one of the first data register and the second data  
19 register in response to an instruction having a condition  
20 predicate;  
21 wherein the annul code comprises a first annul code stored in  
22 the first data register loaded into the annul word memory in  
23 response to the condition predicate being satisfied; and

24 wherein the annul code comprises a second annul code stored in  
25 the second data register loaded into the annul word memory in  
26 response to the condition predicate not being satisfied.

1 18. (Currently Amended) ~~The A processor of claim 1~~ having a  
2 changeable architected state, comprising:  
3 instruction memory for storing instructions;  
4 an instruction pipeline, wherein an instruction which passes  
5 entirely through the pipeline alters the architected state and  
6 wherein the pipeline comprises circuitry for fetching instructions  
7 from the instruction memory into the pipeline;  
8 an annul word memory for storing an annul code having a  
9 plurality of annul bits, each annul bit having a one-to-one  
10 correspondence to one instruction of a group of instructions in the  
11 pipeline; and  
12 circuitry for preventing one or more selected instructions in  
13 the group from altering the architected state in response to the  
14 corresponding annul bit of the annul code;  
15 and further comprising a data register;  
16 wherein the annul code is loaded into the annul word memory  
17 from a selected half of the data register in response to an  
18 instruction having a condition predicate;  
19 wherein the annul code comprises a first annul code stored in  
20 a first one-half of the data register loaded into the annul word  
21 memory in response to the condition predicate being satisfied; and  
22 wherein the annul code comprises a second annul code stored in  
23 a second one-half of the data register different from the first  
24 one-half loaded into the annul word memory in response to the  
25 condition predicate not being satisfied.

Claim 19. (Canceled)

1        20. (Currently Amended) The A processor of claim 1 having a  
2 changeable architected state, comprising:

3        instruction memory for storing instructions;

4        an instruction pipeline, wherein an instruction which passes  
5 entirely through the pipeline alters the architected state and  
6 wherein the pipeline comprises circuitry for fetching instructions  
7 from the instruction memory into the pipeline;

8        an annul word memory for storing an annul code having a  
9 plurality of annul bits, each annul bit having a one-to-one  
10 correspondence to one instruction of a group of instructions in the  
11 pipeline; and

12        circuitry for preventing one or more selected instructions in  
13 the group from altering the architected state in response to the  
14 corresponding annul bit of the annul code;

15        and further comprising a register;

16        wherein the register stores the annul code which comprises a  
17 set of annul bits having a first logical value and a set of annul  
18 bits having a second logical value;

19        wherein the annul code is loaded into the annul word memory  
20 from the register in response to an instruction having a condition  
21 predicate;

22        wherein the circuitry for preventing prevents instructions  
23 corresponding to annul bits having a first logical value from  
24 altering the architected state in response to the condition  
25 predicate being satisfied; and

26        wherein the circuitry for preventing prevents instructions  
27 corresponding to annul bits having a second logical value opposite  
28 the first logical state from altering the architected state in  
29 response to the condition predicate not being satisfied.

21. (Canceled)

1        22. (Currently Amended) A method of data processing  
2 comprising the steps of:

3        identifying at compile time prior to execution a group of  
4 instructions including a tree of a plurality of conditional branch  
5 instructions;

6        at compile time prior to execution for each conditional branch  
7 instruction within the group of instructions

8            forming a first annul code having an annul bit  
9        corresponding to each instruction following the conditional  
10       branch instruction, the annul bit having a first logical state  
11       for instructions following the detected conditional branch  
12       instruction executed if a condition of the conditional branch  
13       instruction is satisfied and a second logical state opposite  
14       to the first logical state for instructions following the  
15       detected conditional branch instruction executed if the  
16       condition of the conditional branch instruction is not  
17       satisfied,

18           forming a second annul code having an annul bit  
19       corresponding to each instruction following the conditional  
20       branch instruction, the annul bit having the second logical  
21       state for instructions following the detected conditional  
22       branch instruction executed if a condition of the conditional  
23       branch instruction is satisfied and the first logical state  
24       for instructions following the detected conditional branch  
25       instruction executed if the condition of the conditional  
26       branch instruction is not satisfied;

27       upon execution of the group of instructions

28           detecting each conditional branch instruction,

29           evaluating the condition of the conditional branch  
30 instruction,

31           loading the corresponding first annul code if the  
32 condition of the conditional branch instruction is satisfied,



33           loading the corresponding second annul code if the  
34           condition of the conditional branch instruction is not  
35           satisfied,  
36           executing each instruction following the conditional  
37           branch instruction if the corresponding annul bit of the  
38           corresponding annul code has the first state, and  
39           not executing each instruction following the conditional  
40           branch instruction if the corresponding annul bit of the  
41           corresponding annul code has the second state.